

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zpracování obrazu pro sportovní a rehabilitační účely

Image Processing for Sports and Rehabilitation Purposes

Zadání bakalářské práce

Student: **Lucie Staňková**

Studijní program: B2649 Elektrotechnika

Studijní obor: 3901R039 Biomedicínský technik

Téma: **Zpracování obrazu pro sportovní a rehabilitační účely**
Image Processing for Sports and Rehabilitation Purposes

Jazyk vypracování: čeština

Zásady pro vypracování:

V rámci bakalářské práce bude student zpracovávat obraz pomocí dvou vhodně umístěných kamer v reálném čase. Úkolem bude detekovat předmět a určit jeho přesnou polohu na definované ploše. Dané místo následně označíte projektorem.

Výsledkem by měla být reálná aplikace psaná v jazyce C# obsluhující kamery a projekci v reálném čase. Tento systém patří mezi moderní sdělovací a informační technologie s vysokým potenciálem vývoje. Zpracování obrazu a následná reakce formou projekce je nový, rychle se rozvíjející trend, který interaguje vizuální formou do všech odvětví průmyslu. V nemocničním prostředí je možné aplikovat na navádění polohy pacienta při vyšetření či jiné oddělení, nevyjímaje ani rehabilitační účely.

Body zadání:

1. Nastudování grafiky a metod ukládání obrazových dat.
2. Teorie zpracování obrazu.
3. Určení pozice předmětu v prostoru za pomoci kamerových systémů.
4. Vytvoření uživatelského prostředí v jazyce C#.
5. Zhodnocení výsledků.

Seznam doporučené odborné literatury:

- [1] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. 1. vyd. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-233-6.
- [2] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Vyd. 1. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-02114-9.
- [3] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing analysis and machine vision*. 2nd ed. Pacific Grove: PWS Publishing, c1999. ISBN 0-534-95393-X.
- [4] SOJKA, Eduard. *Digitální zpracování a analýza obrazů*. 1. vyd. Ostrava: VŠB-Technická univerzita, 2000. ISBN 80-7078-746-5.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Tomáš Urbanczyk**

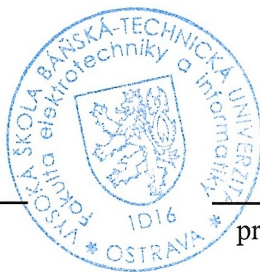
Konzultant bakalářské práce: Ing. Jan Kubiček

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 2018

Blanka Králová
.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 2018

Grasman David
.....

Abstrakt

Obsahem této bakalářské je vytvořením reálné aplikace, v jazyce C#, pro hru squash. Zahrnuje základní metody zpracování obrazu. Aplikace detekuje objekt a určuje jeho polohu na definované ploše. Součástí této práce je vyrobení laserového projektoru a je řešeno jeho ovládání. Je ukázáno, jak je možné tyto poznatky a návrhy využít pro sportovní účely. Taktéž je vyroben návrh pro rehabilitační cvičení pro pacienty po cévní mozkové příhodě.

Klíčová slova: zpracování obrazu, C#, kamera Basler, laserový projektor, rehabilitace, squash

Abstract

This thesis contains implementation of C# application for game of squash. It includes basic methods of image processing. The application can detect an object and determine its position on defined plane. One part of this thesis is focused on building a simple Laser projector with controls. Usability of knowledge for sport purposes is demonstrated. There is also a suggestion for usage of this application for patients after a brain stroke.

Key Words: image processing, C#, video camera Basler, Laser projector, rehabilitation, squash

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
1 Úvod	12
2 Nastudování grafiky a metod ukládání obrazových dat	13
2.1 Digitalizace	13
2.2 Grafické formáty pro ukládání obrazu	14
3 Teorie zpracování obrazu	16
3.1 Barevné modely	16
3.2 Histogram	18
3.3 Metody úpravy obrazu	18
4 Určení pozice předmětu v prostoru za pomoci kamerových systémů	21
4.1 Algoritmy pro vyhledávání objektu	21
4.2 Kamery pro zachycení videa	25
5 IQRF	28
6 Praktické řešení pro sportovní účely	29
6.1 Hřiště na squash	29
6.2 Konstrukce laserového projektoru	30
7 Vytvoření uživatelského prostředí v jazyce C#	35
7.1 GUI	36
7.2 Zahájení snímkování	37
7.3 Zpracování obrazu	38
7.4 Detekce kruhu a ořez vybrané oblasti	40
7.5 Vyhodnocení	41
8 Možné využití v rehabilitaci	42
8.1 Náhle mozkové příhody - fokální cerebrální ischemie	42
8.2 Rehabilitace po cévní mozkové příhodě	42
8.3 Vlastní návrh použití této aplikace pro rehabilitaci	43
9 Zhodnocení výsledků	45

10 Závěr	46
Literatura	47

Seznam použitých zkratek a symbolů

CCD	– Charge-Coupled Device
CMOS	– Complementary Metal-Oxide-Semiconductor
TIFF	– Tagged Image File Format
PNG	– Portable Networks Graphics
MNG	– Multiple-image Network Graphics
JPEG	– Joint Photographic Experts
BMP	– Bitmap
RGB	– barevný model Red, Green, Blue
CMY, CMYK	– barevný model Cyan, Magenta, Yellow, Key
HSV	– barevný model Hue, Saturation, Value
ROI	– Region of interest
PCA	– Principal COmponent Analysis
PC	– statická komponenta
TSS	– vyhledávání centroidu ve vyhledávacím okně
SIM	– Subscriber Identity Module
VCC	– napájení
GND	– uzemnění

Seznam obrázků

1	RGB model krychle [8]	16
2	CMY model krychle [8]	17
3	HSV barevný model [8]	17
4	Histogram vytvořený v grafickém editoru	18
5	Konvoluční jádra nejvýznamějších hranových detektorů, upraveno [20]	19
6	Algoritmus PN - upraveno [1]	22
7	Diagram CamShift algoritmu - upraveno [9]	23
8	DP-CamShift algoritmus - upraveno [9]	24
9	Basler acA2000-165uc	26
10	Objektiv	26
11	Akční kamera	27
12	IQRF: vlevo CK-USB-04A, vpravo DK-EVAL-04A	28
13	Schéma SIM konektoru TR-72D - upraveno [4]	28
14	Kurt s náčrtem rozmístění techniky	29
15	Jednotlivé mechanické součásti systému [2]	30
16	Zrcátková část laserového projektoru	30
17	Náčrt projekce	31
18	Úhel mezi zrcátkem a chladičem	31
19	Blokové schéma připojení IQRF se servomotory, upraveno [4]	32
20	Ukázka GUI na ovládání IQRF	34
21	Diagram aplikace	35
22	Uživatelské rozhraní	36
23	Snímek z kamery Basler bez úprav	38
24	Snímek z kamery Basler po převedení do HSV barevného modulu	39
25	Snímek z kamery Basler po převedení do HSV barevného modulu a následně převeden do monochromatického obrázku	39
26	Snímek z kamery Basler po převedení do HSV barevného modulu, následně do monochromatického obrázku a aplikován Cannyho hranový detektor	40
27	Box and Block Test, upraveno dle [18]	42
28	Nine Hole Peg Test [6]	43
29	Nákres pro rehabilitační cvičení po mozkové příhodě	44

Seznam tabulek

1	Parametry kamery Basler	26
2	Parametry akční kamery	27

1 Úvod

Tato bakalářská práce popisuje základní metody ukládání obrazových dat, zpracování obrazu a shrnuje nejčastěji používané obrazové grafické formáty. Objasňuje postup digitalizace obrazu a základní barevné modely. Níže popsané úpravy obrazu se realizují pomocí jednoduchých obrazových transformací. Takové transformace mohou být jasové, či převody do odstínu šedi. Dále sumarizuje základní algoritmy pro extrakci objektu, či určení jeho polohy.

V práci je popis výroby jednoduchého laserového projektoru. Projektor je zkonstruován za pomoci zrcátka, větráčku z počítače, vytisknutých 3D modelů, laseru, servomotorů a IQRF. Díky rotaci větráčku s připevněným zrcátkem je projektor schopen promítnut na zeď jednoduchý kruh a s pomocí servomotorů a modulu IQRF je umožněno, aby se promítaný kruh pohyboval po ose x a y .

Následně jsou tyto znalosti aplikovány pro reálnou aplikaci, která obsluhuje kameru a projektor a vyhodnocuje výsledky této aplikace. Celá aplikace je napsána v jazyce C# a využívá knihoven EmguCV.

Dále je navrženo použití této aplikace pro sportovní účely. V této práci to je sport squash. Aplikace umožňuje sledovat progres hráče. Hra spočívá v počítání úspěšných zásahu míčkem do promítaného kruhu. Průměr kruhu se dá zvětšovat a zmenšovat dle aktuálních potřeb hráče.

Na závěr této práce je navrženo použití pro rehabilitaci pacientů po cévní mozkové příhodě.

2 Nastudování grafiky a metod ukládání obrazových dat

Aby bylo možné 3D obraz zachytit a zobrazit, je nutné tento obraz zmapovat jako dvourozměrný. Takový obraz se nazývá digitální a vytvoří se kvantováním, kdy hodnota spojitě funkce je rozdělena na diskrétní hodnoty a prahováním. Obraz se skládá z prvků (pixelů), z nichž má každý určitou polohu (x, y) a určitou hodnotu jasu, či šedou úroveň.

2.1 Digitalizace

Technické prostředky nám většinou neumožňují zachytit video a uložit ho jako spojitou informaci. Obraz musíme převést na digitální signál, který už počítač uložit dokáže. Takový signál má diskrétní hodnoty a provádí se vzorkováním a kvantováním.

[5]

2.1.1 Vzorkování

Vzorkování vzniká při snímání hodnot ze spojitě funkce v daném vzorkovacím intervalu Δx . Je zvolena vzorkovací frekvence, která odpovídá převrácené hodnotě jednotlivých vzorků. Tato frekvence musí splňovat tzv. Shannonův teorém, který říká, že vzorkovací frekvence musí být dvojnásobná než maximální frekvence signálu.

$$f_{max} < 1/2 * \Delta x \quad (1)$$

U digitálních fotoaparátů a kamer je jemnost vzorkování dána počtem aktivním bodů senzoru - CCD nebo CMOS.

[5]

2.1.2 Kvantování

Kvantování je proces při kterém je naměřené veličině přiřazena diskrétní hodnota pomocí A/D převodníku. Hodnota jednotlivých úrovní N je udávána pomocí bitů dle vzorce

$$N = 2^K, \quad (2)$$

kde K je potřebný počet bitů. Běžně se používá 8 bitů = 256 kvantizačních úrovní. Při kvantování dochází ke ztrátě informace a vzniká kvantizační chyba, kterou je možno eliminovat zvýšením počtu kvantizačních úrovní.

[5]

2.2 Grafické formáty pro ukládání obrazu

Formáty pro ukládání mohou být rastrové, nebo vektorové. Digitální fotoaparáty používají rastrové formáty. Tento rastrový obraz může být kompresovaný bezztrátově, nebo ztrátově. Kompresi bezztrátová znamená, že po jeho uložení lze soubor otevřít opět jako originální obrázek. Takovou kompresi podporují formáty TIFF, RAW, PNG, MNG. Ztrátovou kompresi používá formát JPEG.

[5]

2.2.1 TIFF - Tag Image File Format

Tento formát není moc používaný, protože má velký objem ukládaných dat, a složitý popis formátu. Obrázek je ukládán zcela beze ztrát. Takto uložený obrázek je vhodný pro další zpracování a úpravu.

[5]

2.2.2 RAW

U formátu RAW se zapisují hodnoty ze snímače přesně tak, jak byly naměřeny. Převod do RGB se vypočítává až v počítači příslušným softwarem. U tohoto formátu jsou data ukládána s takovou barevnou hloubkou, jakou je zachytil snímač. Ostatní formáty podporují 8 nebo 16 bitové kódování.

[5]

2.2.3 PNG - Portable Network Graphics

PNG umožňuje ukládat data i v 48 bitové barevné hloubce, má bezztrátovou kompresi, takže vlivem komprese nedojde ke ztrátě detailů v obraze. Umožňuje i zahrnutí některých nastavení jako je použití informace o průhlednosti, či nastavení gama korekce.

[5]

2.2.4 JPEG - Joint Photographic Expert Group

Každým uložením snímku dochází ke ztrátě dat dle nastavené komprese. Kompresi je výpočetně náročná. Využívá jasové separace a separaci barvonosné složky obrazu, na který je aplikována diskrétní kosinová transformace. Tato transformace je založen na diskrétní Fourierově transformaci. Nachází se zde sada parametrů, jejichž skládáním se rekonstruuje původní obraz. U tohoto formátu je možno nastavit různou míru kompresi. Pokud zvolíme vysokou kompresi, dojde k vyšší ztrátě původní informace.

[5, 8]

2.2.5 MNG

Tento formát je vytvořený pro webové animace. Může mít až 16 milionů barev. Ovšem k jeho fungování je potřeba nainstalovat ve webovém prohlížeči patřičný plug-in.

[5]

2.2.6 BMP

BMP neboli bitmap formát slouží k ukládání rastrové grafiky, formát se ukládá po jednotlivých pixelech. Každý pixel odpovídá jednomu nebo více bitům v bitmapě. U jednobarevného obrázku odpovídá 1 pixel 1 bitu, u barevných obrázků se používá barevná mapa 256 barev a to vyžaduje 8 bitů na pixel.

[13]

3 Teorie zpracování obrazu

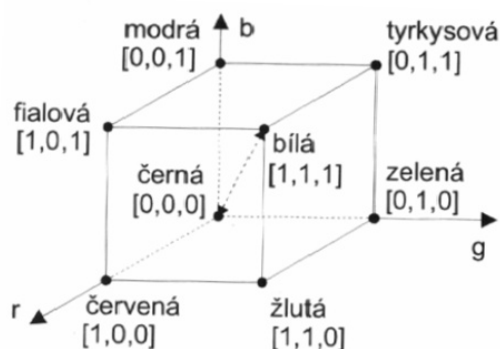
V této kapitole jsou popsány barevné modely obrazu a jednoduché metody úpravy obrazu.

3.1 Barevné modely

Barevný obraz je reprezentován pomocí matice hodnot, která značí jednotlivou sytost základní barvy. Mísení těchto barev probíhá podle různých modelů, některé jsou zmíněné níže. Takové mísení může být buď aditivní, kdy se jednotlivé barevné složky sčítají a nebo subtraktivní, kdy se jednotlivé barevné složky se odčítají.

3.1.1 RGB

RGB je z anglického Red, Green, Blue a vytváří se aditivním složením třech barev - červené, zelené a modré. Konkrétní hodnoty jednotlivých barevných kanálů jsou určeny vlnovými délkami 700 nm pro červenou, 546,1 nm pro zelenou a 435,8 nm pro modrou. Souřadnice počátku na krychlovém modelu (obr. 1) odpovídají barvě bílé $[1,1,1]$ a vrchol krychle reprezentuje barvu černou $[0,0,0]$.



Obrázek 1: RGB model krychle [8]

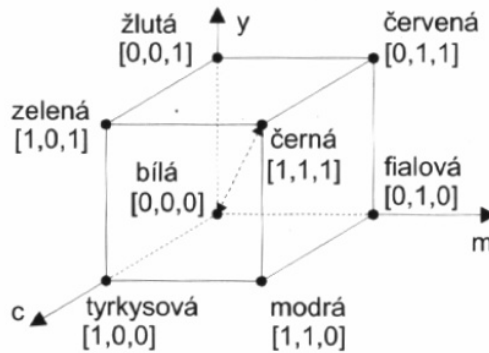
Tento barevný model se používá nejčastěji při snímání barevného obrazu a následném zobrazování na monitoru. Různé grafické softwary také využívají model RGBA, kde je navíc přidána informace o průhlednosti. Tento kanál se nazývá alfa kanál a značí se A.

Aditivním mícháním těchto barev získáváme výslednou barvu, která je vyjádřena počtem bitů, kde každá jednotlivá barva může být reprezentována obvykle 8 bity. Čím vyšší hodnoty dané barvy, tím je výsledná barva světlejší.

[5]

3.1.2 CMY, CMYK

Tento model je tvořen subtraktivním mícháním barev Cyan (azurová), Magenta (růžová) a Yellow (žlutá). Míchání těchto barev se využívá převážně pro tisk.



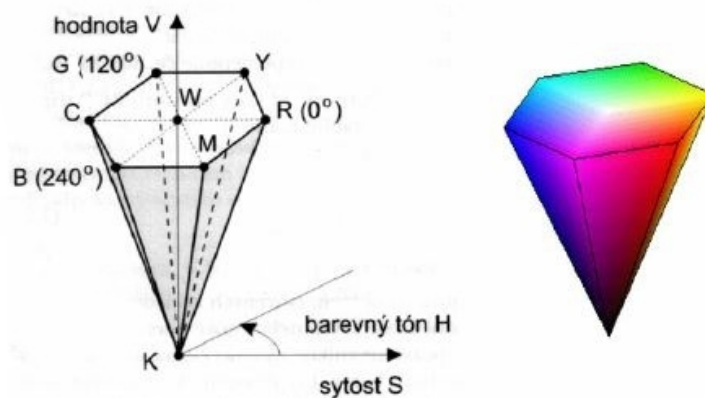
Obrázek 2: CMY model krychle [8]

Souřadnice počátku na krychlovém modelu (obr. 2) odpovídají barvě bílé $[0,0,0]$ a vrchol barvě černé $[1,1,1]$.

[5]

3.1.3 HSV

Výsledná barva je tvořena pomocí složek Hue (barevný tón), Saturation (sytylost) a Value (hodnota jasu). Takový barevný model je znázorňován nikoliv krychlí, ale pomocí jehlanu (viz obr. 3).



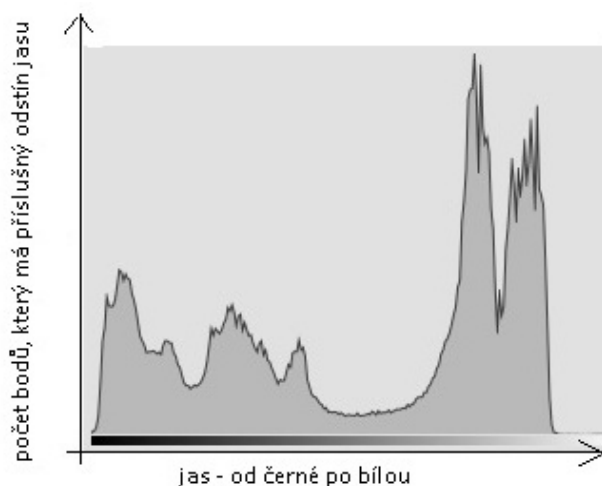
Obrázek 3: HSV barevný model [8]

[8]

3.2 Histogram

Histogram představuje globální rozložení jasových úrovní v obraze. Jednotlivým obrazům je přiřazován jiný histogram. Na vodorovnou osu grafu histogramu se vynášejí jasové úrovně a na svislou osu četnost bodů, který má příslušný odstín jasu. Často se zobrazuje jako sloupcový diagram (viz obr. 4).

[7]



Obrázek 4: Histogram vytvořený v grafickém editoru

3.3 Metody úpravy obrazu

Aby bylo možné obraz upravovat či zdůraznit určité objekty v obraze, je potřeba obraz transformovat. Mezi základní transformace patří převody barevného obrazu do různých jiných odstínů a jasová transformace.

3.3.1 Převod barevného obrazu do monochromatického obrazu

Barevný obraz se dá převést na monochromatický obraz tak, že jednotlivé monochromatické složky vyjádříme hodnotami jasu. Jednotlivé jasové úrovně se vypočítají jako:

$$f = 0,299R + 0,587G + 0,114B, \quad (3)$$

kde f je výsledná úroveň jasu v šedotónovém obraze. Hodnoty R , G , B obsahují úrovně jasů jednotlivých pixelů.

[5]

3.3.2 Segmentace obrazu prahováním

Každá oblast obrazu je charakterizována odrazivostí a pohltivostí povrchu. Tudíž objekt a pozadí mají rozdílné vlastnosti. Tento práh se určuje z histogramu a to buď procentuálně, nebo z tvaru histogramu.

[8]

3.3.3 Detekce hran

Tato detekce je založena na metodách využívající první derivaci, nebo druhou derivaci.

Detekce pomocí první derivace se získává jako rozdíl pixelů v obrazu a vypočítává se derivace pro řádky, resp. pro sloupce. Výpočet výsledného gradientu je podle rovnice:

$$G(i, j) = \sqrt{G_R(i, j)^2 + G_S(i, j)^2} \quad (4)$$

Jsou různé typy těchto detektorů. Tyto typy se liší dle matic (masek), podle které se provádí výpočet derivace. Příklady takových matic je uvedeno v obr. 5.

Operátor	Gradient řádků	Sloupcový gradient
Pixelový rozdíl	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Robertsův operátor	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Sobelův operátor	$\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

Obrázek 5: Konvoluční jádra nejvýznamějších hranových detektorů, upraveno [20]

Při detekce pomocí druhé derivace se vyžívá průchod nulou. Tyto detektory jsou nevýhodné z důvodu, že při derivaci detekující průchod nulou je velké vyhlazení obrazu, tudíž vzniká ztráta ostrých rohů se sklonem vytvářet uzavřené smyčky hran. Představitel detektoru hran využívající detekci hran podle druhé derivace je Laplacian, který detekuje hrany ve všech směrech.

[20]

Cannyho hranový detektor je jeden z nejpřesnějších z výše zmiňovaných hranových detektorů. Je navržen tak, aby co nejlépe splňoval detekci, lokalizaci a jedinečnou odezvu. Stručný popis fungování tohoto detektoru, za předpokladu, že obrázek je ve stupních šedi a je eliminován šum, je následující:

1. provede se standardní detekce hran dle Sobelova operátoru a vypočte se směr a velikost gradientu.

2. naleznou se lokální maxima

3. je aplikováno prahování. Tento práh má dvě hodnoty - maximální práh a práh minimální. Tyto prahy umožňují dodatečnou eliminaci šumu. Hodnoty, které jsou vyšší než horní práh jsou právoplatné body hrany a pixely, které s těmito body sousedí a jejichž gradient je vyšší než dolní práh, jsou taktéž právoplatnými body. Tento postup umožňuje odstranit šum a nevýznamné hrany. Tyto nevýznamné hrany totiž způsobují detekci falešných hran.

[11]

3.3.4 Jasová transformace

Jasová transformace je nejjednodušší způsob vylepšení vzhledu obrázku. Tato transformace se provádí buď ve frekvenční nebo prostorové doméně. Při frekvenční doméně se obraz transformuje pomocí Fourierovy transformace. V prostorové doméně se na jednotlivé pixely aplikuje úroveň jasu. Neuvažuje se okolí zpracovaného bodu, ale pouze jeden pixel. Původní obraz s hodnotami jasu $f(x, y)$ se transformuje jako nový obraz s hodnotami $g(x, y)$.

Vztah pro prostorovou doménu:

$$g(x, y) = T[f(x, y)], \quad (5)$$

kde T je transformace. Hodnota jasu je v intervalu od 0 do 1, kde hodnota 0 je černá a hodnota 1 odpovídá barvě bílé. Mezi takové transformace patří gama korekce, negativ, prahování a roztažení kontrastu.

Negativ - intenzita jasu každého bodu je nahrazena novou hodnotou, stranově převrácenou $g(x, y) = 1 - f(x, y)$, pro všechny x, y v obraze.

Gama korekce - $g = c * f^\gamma$, kde c a γ je konstanta a f je jasový signál. Používá se při korekci snímku za účelem zviditelnění některých tmavých částí.

[5]

4 Určení pozice předmětu v prostoru za pomoci kamerových systémů

Odborných článků pro detekování objektů z videa je nespočet. V následujícím textu jsou shrnuty ty nejčastěji používané.

4.1 Algoritmy pro vyhledávání objektu

Pro zpracování této bakalářské práce je potřeba najít vhodný algoritmus, tedy teoretický princip řešení, pro určení pozice a vyhledávání objektů. Takové algoritmy mohou být následující.

4.1.1 ROI (Region of interest) metoda v kombinaci s PN algoritmem

Tato metoda používá PCA algoritmus, detekci objektu a PN algoritmus.

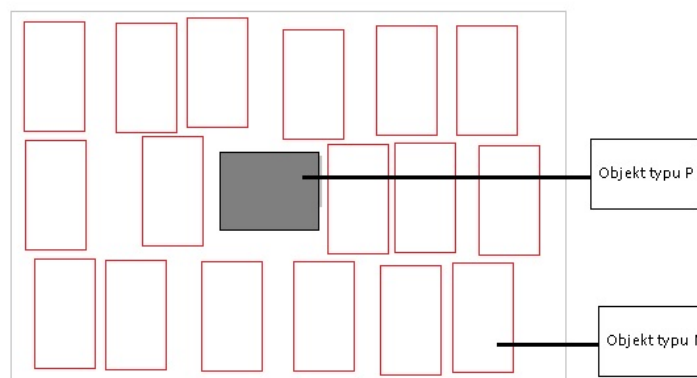
PCA (Principal Component Analysis) je statistická analýza dat, která se využívá pro zkrácení doby potřebné k výpočtu. Vytváří se hlavní komponenta (PC), která obsahuje jak statistické tak dynamické informace o sekvenci obrazu. Tyto objekty se extrahují v každém kroku. Každý snímek obsahuje tři složky (R, G, B) a tyto složky jsou sestaveny tak, aby vytvořily sadu jednopásmových obrázků. PCA se provádí na každé sadě snímků a produkuje PC pro každé jednotlivé pásmo. Nakonec jsou všechny části spojeny a vzniká PC sekvence.

Algoritmus metody PCA je následující:

1. Je zachyceno video ze systému
2. Je vybrán objekt a oblast kolem něj je oříznutá
3. Vytvoří se prázdné obrazové pole
4. Oříznutý objekt je uložen do matice s indexem 0
5. Je vytvořen ROI ve vzdálenosti 20 pixelů
6. Pro zjištění umístění objektu je použit algoritmus PN
7. Je použito PCA pro sledování pohybujícího se objektu spolu s detektorem objektů a je sledována shodnost snímků v procentech
8. Pokud není nalezena shoda mezi snímky, detekují se chyby pomocí PN algoritmu

Celý PN algoritmus pak spočívá v tom, že uživatel vybere objekt, který má být sledován. Takový objekt se nazývá typ P. Pozadí za objektem je rozděleno na typy N. Objekt P se stává referenční šablonou. Každý snímek se procentuálně porovnává se šablonou, pokud je shodný s typem objektu N, pak je obraz zanedbán, protože došlo k falešnému zjištění. Pokud je shodný s P typem, pak je snímek uložen do pole. PN algoritmus je na obr. 6.

[1]



Obrázek 6: Algoritmus PN - upraveno [1]

4.1.2 Meanshift

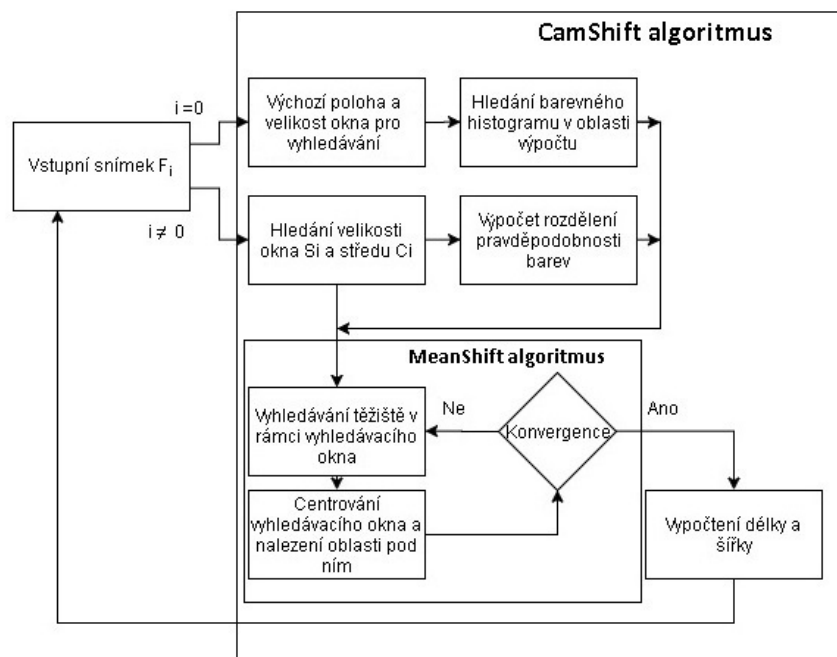
Je to iterativní (opakované užití výstupu z jedné iterace ke vstupu do iterace následující) optimalizační algoritmus. Je uvažována distribuce pixelů. Vytvoří se počáteční okno a určí se střed. Pokud je však v obraze nalezen centroid, pak je tento centroid skutečným centrem okna. Přemísťuje se okno tak, aby nové okno odpovídalo předchozím centroidům. Znova se vyhledá nový centroid. Takto vyhledávání pokračuje, dokud tento střed nebude vyhledávat stále stejné místo a toto okno je maximální hustota distribuce pixelů. Tedy se hledají body, které konvergují do stejného lokálního maxima.

[16]

4.1.3 CamShift

Nevýhoda u výše zmíněné metody MeanShift je ta, že vyhledávací okno má stále stejnou velikost a pokud je objekt v pohybu, musí být i vyhledávací okno přizpůsobené rotaci a velikosti objektu. Metoda CamShift upravuje tuto velikost okna a to pokaždé, když MeanShift konverguje. Zároveň je vypočítána orientace. Znova se použije MeanShift s novými parametry vyhledávání s předchozím umístěním okna. Tento proces se opakuje, dokud není splněna požadovaná přesnost. Tento diagram je zobrazen na obrázku 7.

[16]

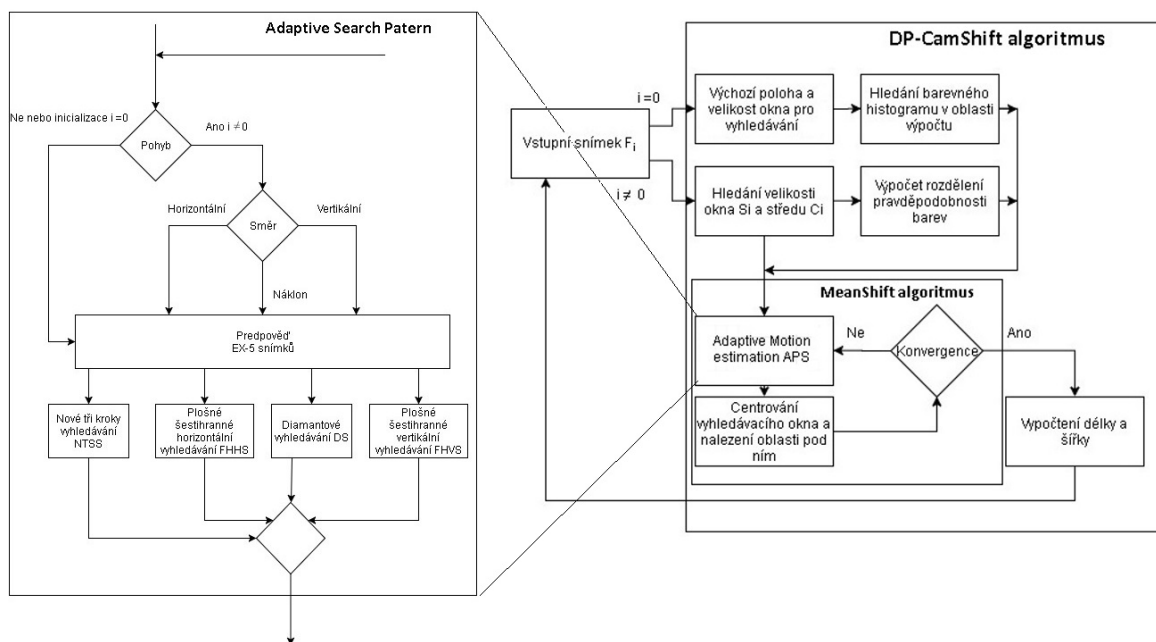


Obrázek 7: Diagram CamShift algoritmu - upraveno [9]

4.1.4 DP-CamShift

Vzhledem k přizpůsobivému přístupu CamShift algoritmu, může být nahrazeno TSS (vyhledávání centroidu ve vyhledávacím okně) ASP modulem. ASP modul dokáže adaptivně přiřadit správnou ME metodu (odhad pohybu) iterativnímu jádru pro daný směr pohybu sledovaného objektu. ASP modul vylepšuje přesnost předpokládaného směru pohybu.

Celý algoritmus s ASP modulem DP-CamShift vypadá viz obrázek 8.



Obrázek 8: DP-CamShift algoritmus - upraveno [9]

[9]

4.1.5 Detekce založená na tvaru objektu

Objekt se detekuje na základě aproximace kontur. Procesem snižování počtu bodů na křivce získáme aproximovanou verzi. Pokud má aproximovaná kontura objektu tři vrcholy, bude to trojúhelník. Pokud má čtyři vrcholy, je to buď čtverec, nebo obdélník. Zde se tvar určuje podle poměru stran tvaru, což je šířka obdélníku, vydělená výškou. Pokud tento poměr bude 1,0 je to čtverec, v opačném případě má tvar obdélníku. Pokud má obrys pět vrcholů výsledný tvar je pentagon. Když objekt nebude ani jeden zmíněný, můžeme předpokládat, že zkoumaný tvar je kruh.

[17]

4.1.6 Barevná extrakce

Pro detekci objektů se používá algoritmus, který extrahuje jednotlivé barvy. Hledaný objekt má unikátní barvu, která je odlišná barvy od barev ostatních objektů. Vyhledává se pomocí jednotlivých linek obrazu, na které je obraz rozdělen. Tato metoda tak zpracovává jednotlivé linky obsahující dané pixely a po vyhodnocení se přesouvá k lince další. Výsledkem jsou hodnoty pixelů v barevném prostoru spolu se souřadnicemi objektů. Tato detekce je oproti tvarové detekci mnohem efektivnější a rychlejší.

[15]

4.1.7 Metoda odčítání pozadí

Je to velmi účinný algoritmus, který se používá pro vyhledávání pohyblivých objektů ve snímcích. Je používán u statických kamer. Tato metoda je založena na odčítání jednotlivých pixelů z aktuálního snímku od jednotlivých pixelů referenčního snímku $P[B]$.

$$P[F(t)] = P[I(t)] - P[B] \quad (6)$$

$P[F(t)]$ je video rámeček popředí a $P[I(t)]$ je vstupní rámeček pro každou iteraci. Poté je tento odečtený obraz segmentován prahováním, kdy se bodům pozadí přiřadí hodnota 0 a ostatním bodům (bodům popředí) se přiřadí 1. Hodnota 1 odpovídá černé barvě a hodnota 0 reprezentuje barvu bílou. Dále se aplikuje Kalmanův filtr, který odstraňuje, za pomoci odhadu neznámých proměnných, nepřesnosti způsobené šumem.

[14]

4.2 Kamery pro zachycení videa

Pro zpracování bakalářské práce je potřeba zachytit obraz, squashový míček a následně tato data zpracovat. Protože nejvyšší naměřená hodnota (světový rekord) je 281,6 km/h, je zapotřebí kamera, která zvládne vyšší snímkovací frekvenci (fps). Ideální vypočtená snímkovací frekvence, pro kvalitní záznam, je 70 fps a více. Z tohoto důvodu je nutností mít k dispozici sofistikovanější kameru, která tuto snímkovací frekvenci umožňuje. Takovou kamerou je například kamera od firmy Basler, která byla pro tuto práci využita.

Aby kamery dokázaly zachytit obraz, potřebují mít čip, který toto snímání umožňuje. Nejčastěji používané čipy snímače kamer jsou CCD a CMOS.

CCD čipy se skládají z mnoha světlocitlivých buněk, které produkují elektrický náboj, jehož velikost je úměrná dopadu světla. Obsahují registr, který snímá data po řádcích, které se přes zesilovač a A/D převodník digitalizují. Takto registr pokračuje řádek po řádku.

Čipy CMOS jsou vyráběny jednoduchou technologií. Tyto čipy se dělí na pasivní a aktivní. Pasivní čipy se skládají pouze z fotodiod. Aktivní čipy zesilují každou buňku a obsahují obvod odstraňující šum. Tyto čipy mají nižší šum, než čipy pasivní.

Hlavní rozdíl mezi jednotlivými čipy je jejich spotřeba - CMOS čipy mají spotřebu nižší, ale vyšší šum oproti čipu CCD, který má obraz kvalitnější.

[21]

4.2.1 Kamera Basler acA2000-165uc

Kamera Basler acA2000—165uc (obr.9) je kamera s barevným 2 Mega pixelovým čipem CMV2000 – 2E5C. Čip je schopen zachytit až 165 snímku za sekundu o velikosti 2040 x 1086 pixelů. Tuto kameru je možno připojit k počítači USB Micro-B konektorem. Další specifikace kamery jsou uvedeny v tabulce 1.



Obrázek 9: Basler acA2000–165uc

Tabulka 1: Parametry kamery Basler

Parametr	Hodnota
Rozlišení	2040 x 1086
Senzor	CMOSIS CMV2000-2E5C Progresivní skenování CMOS
Velikost pixelů (šířka x výška)	5,5 μm x 5,5 μm
Snímkovací frekvence	až 165 fps
Barevný obraz	ANO
Napájení kamery	Nominální napětí 5V DC dodávaní přes port USB 3.0
Velikost kamery	29,3 mm x 29 mm x 29 mm

Ke kameře je možnost zvolení různých typů objektivů. Používají se hlavně objektivy s pevnou ohniskovou vzdáleností.

V této práci byl použit objektiv(obr.10) s ohniskovou vzdáleností f5 mm a světelností F1,4.

[3]



Obrázek 10: Objektiv

4.2.2 HD Akční kamera 1080P

Jako druhou kameru, která byla využita pro práci k zachycení pohybu člověka, je HD Akční kamera 1080P. Má k dispozici širokoúhlý objektiv 120°. U této kamery se dá nastavit vyvážení bílé, expozice i ISO a takový záznam se může uložit na paměťové médium (Micro SDHC kartu), nebo se kamera dá pomocí USB 2.0 připojit k PC.



Obrázek 11: Akční kamera

Základní parametry jsou vypsány v tabulce 2.

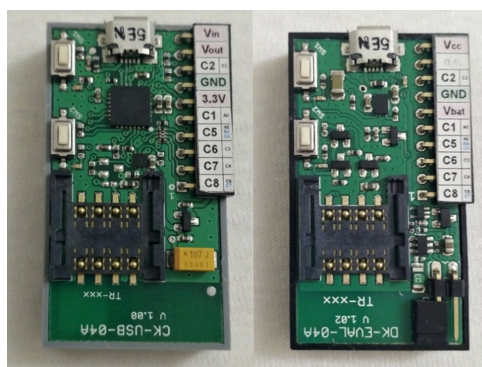
Tabulka 2: Parametry akční kamery

Parametr	Hodnota
Rozlišení	1920 x 1080
Senzor	12 mega pixelový CMOS senzor
Zoom	4x (digitální)
Snímkovací frekvence	30 fps
Napájení kamery	900mAh Lithiová baterie
Velikost kamery	24.6 x 59.3 x 41.1mm

5 IQRF

IQRF je platforma (obr. 12), která se může používat s jakýmkoli elektronickým zařízením při telemetrii, průmyslovém řízení a automatizaci budov. Komunikace probíhá pomocí bezdrátového přenosu a jeho dosah jsou desítky až stovky metrů. Využití této platformy je vhodné pro svou jednoduchou implementaci.

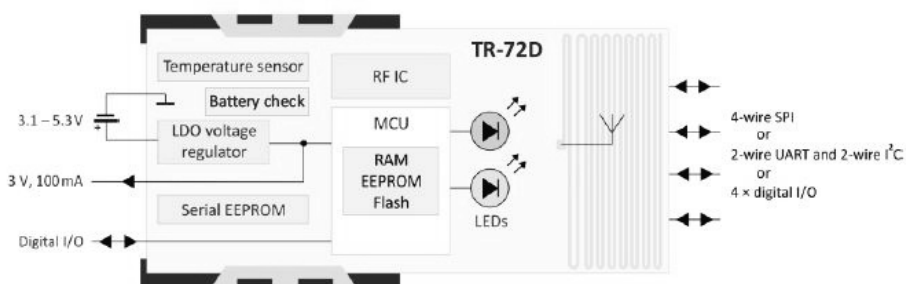
Ve své práci ho využívám k ovládání serva, které pohybuje konstrukcí se zrcátkem.



Obrázek 12: IQRF: vlevo CK-USB-04A, vpravo DK-EVAL-04A

IQRF se skládá z bezdrátového RF přijímače a má svůj vlastní operační systém. Tato platforma je založena na mikročipu PIC16(L)F1938/9. Samotné programování probíhá pomocí IQRF IDE4. Bezdrátový přenos je uskutečňován pomocí vysílačů o velikosti SIM karty. Schéma takového vysílače je na obrázku 13.

[4]



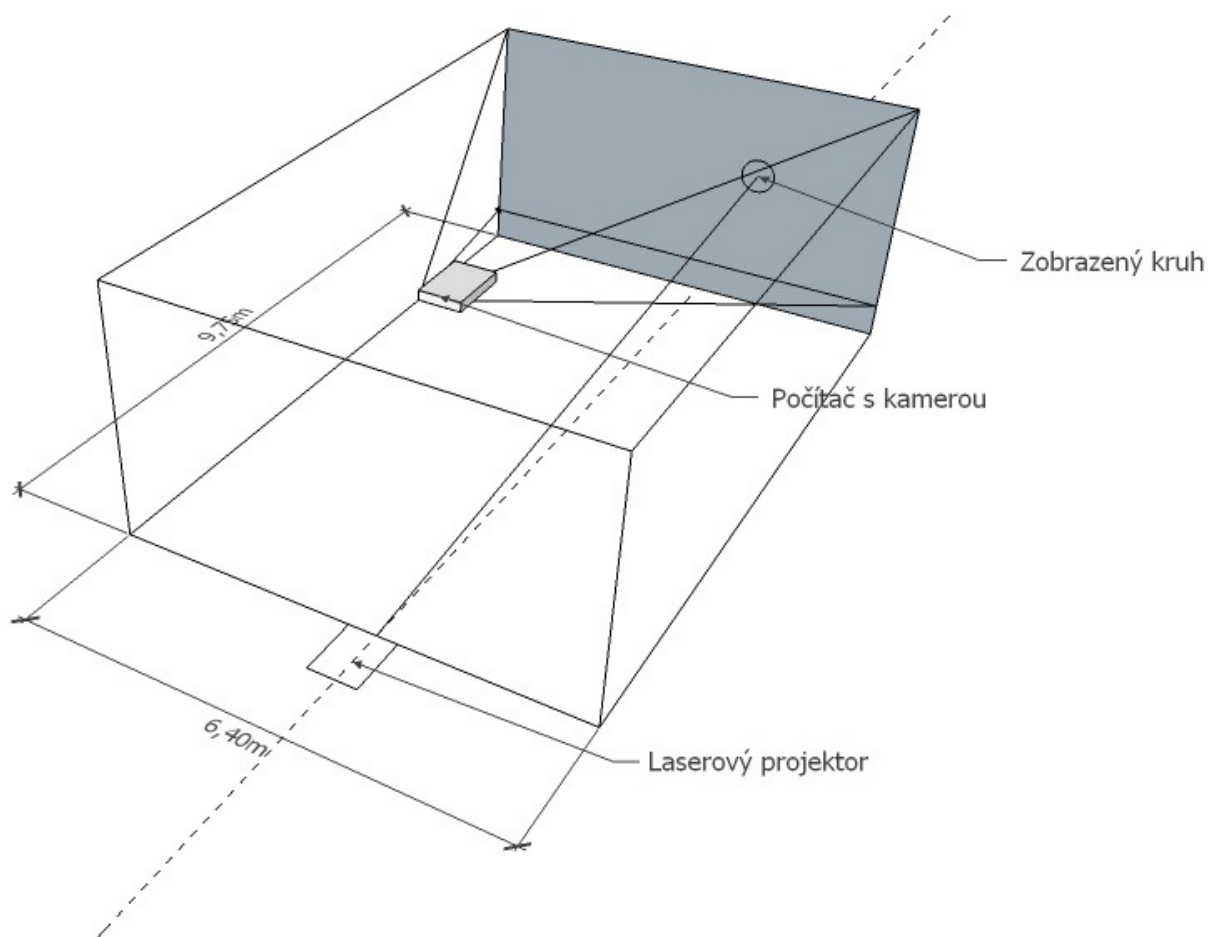
Obrázek 13: Schéma SIM konektoru TR-72D - upraveno [4]

6 Praktické řešení pro sportovní účely

Celý program je založen na počítání zásahu míčku do promítaného kruhu při hře squash. Tento kruh je promítán za pomoci laserového projektoru vlastní výroby. Obsahuje laser, zrcátko, polohovací konstrukci, servomotory a modul IQRF. Hrací plocha je snímána vysokorychlostní kamerou Basler. Po zpracování obrazu lze ve snímku detekovat kruh, který je sledován a aplikace vyhodnocuje, zda-li se hráč do tohoto kruhu trefil míčem. Pokud je míč v tomto kruhu detekován, je tato detekce zaznamenána a v uživatelském rozhraní se objeví text „Detekce míče“.

6.1 Hřiště na squash

Kurt squashe má standardizované rozměry 9,75 m x 6,40 m. Přední stěna je rozdělena čarou ve výšce 48 cm a nad touto čarou se nachází hrací plocha. Protilehlá stěna je prosklená s dveřmi. Za touto prosklenou stěnou je umístěn laserový projektor. Zhruba ve 2/3 délky tohoto hřiště je umístěn notebook spolu s videokamerou. Tento notebook je připevněn ke stropu, aby neomezoval pohyb hráče (viz obr. 14).



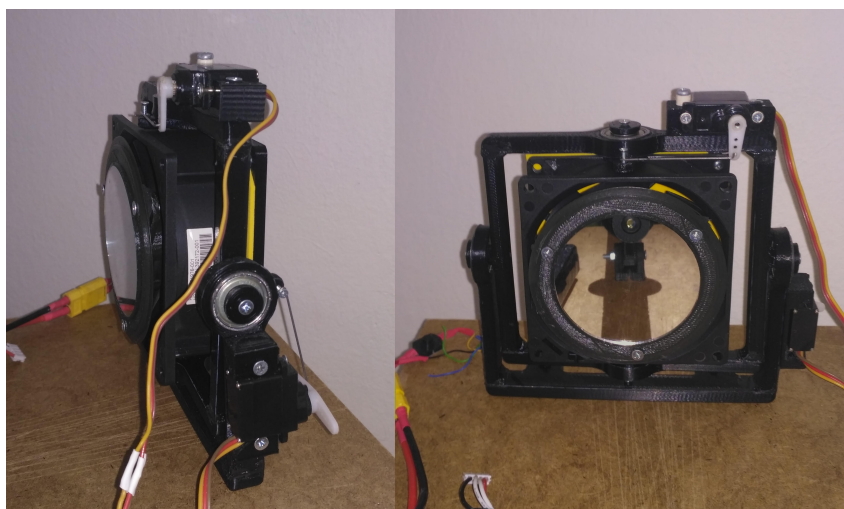
Obrázek 14: Kurt s náčrtem rozmístění techniky

6.2 Konstrukce laserového projektoru

Laserový projektor je složen z výkonného laseru o vlnové délce 532 nm (barva zelená), jehož výkon je udáván jako 5 mW. Dále je ke konstrukci potřeba zrcátko, které je připevněno k ventilátoru z počítače a tento ventilátor je připojen k baterii, která jím otáčí. Laser míří na zrcátko, které díky odrazu a rotaci promítá kruh. Celý mechanismus je usazen v plastové konstrukci, jehož nákres (obr. 15) je k dispozici na serveru thingiverse.com a následně vytisknut 3D tiskárnou. Tato konstrukce je otáčena pomocí dvou servomotorů, které tuto konstrukci naklánějí po ose x a y. Výsledný laserový projektor je na obrázku 16.

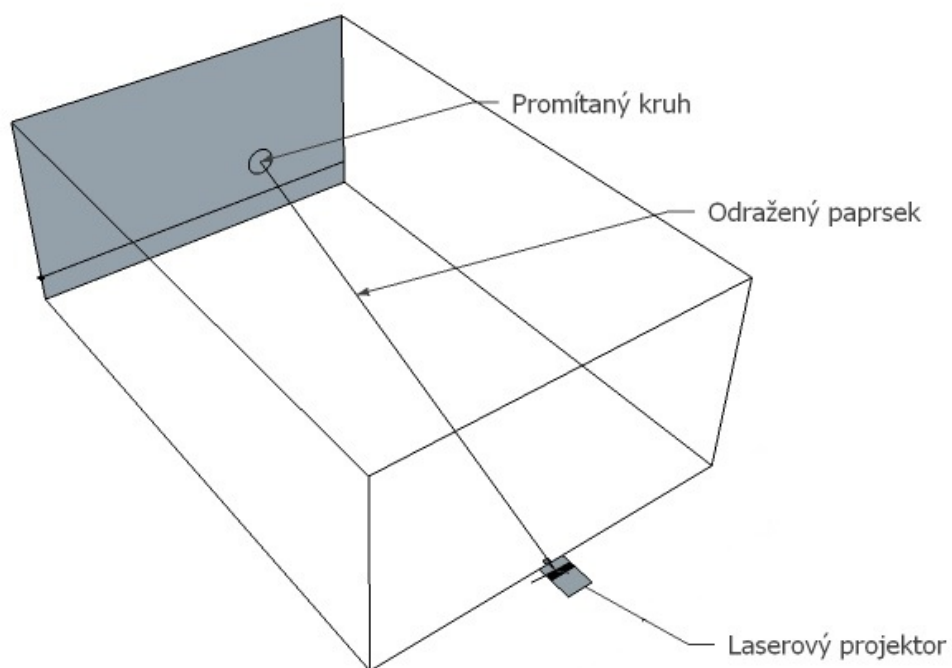


Obrázek 15: Jednotlivé mechanické součásti systému [2]

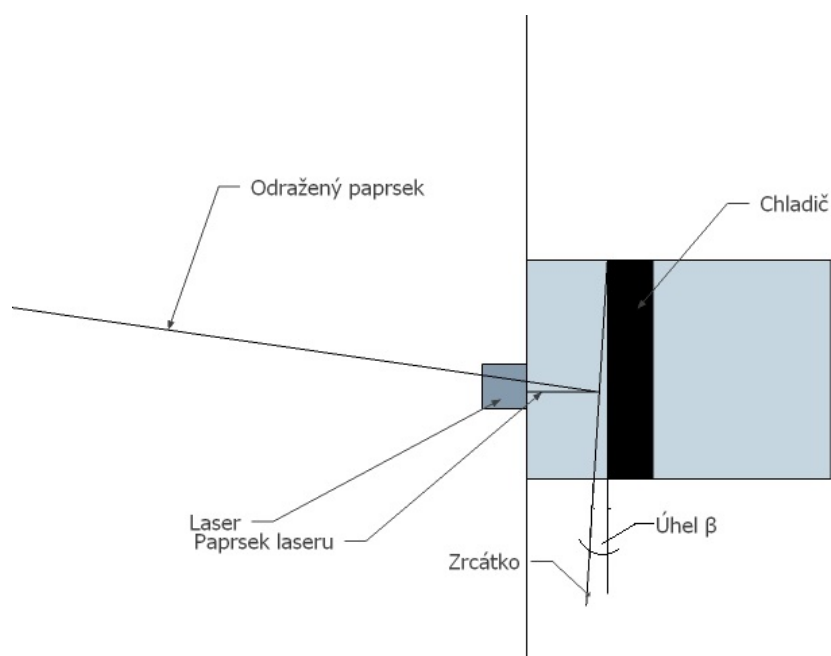


Obrázek 16: Zrcátková část laserového projektoru

Aby bylo možno promítat na zeď kruh o určitém poloměru (viz obr. 17), je zapotřebí vypočítat úhel, který svírá chladič a zrcátko, a tento úhel β následně nastavit pomocí šroubků. Náčrt úhlu naklonění je na obrázku 18.



Obrázek 17: Náčrt projekce

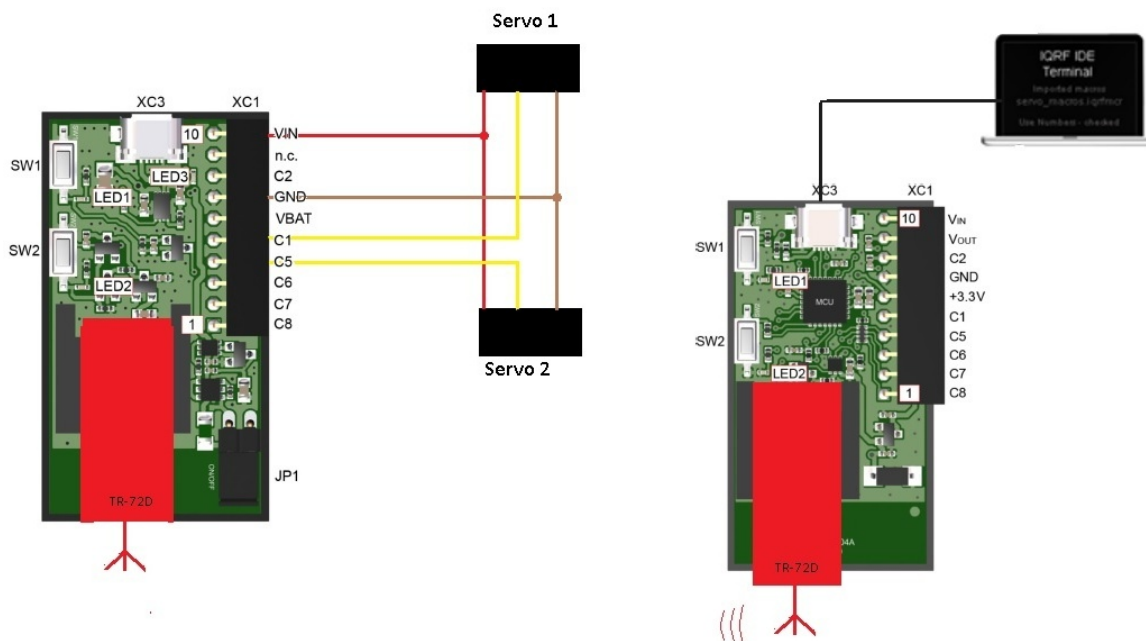


Obrázek 18: Úhel mezi zrcátkem a chladičem

6.2.1 Ovládání laserového projektoru po hrací ploše

Aby bylo umožněno polohu promítaného kruhu měnit, je zapotřebí, aby se zrcátková část naklápěla po ose X a Y a tím měnit úhel dopadajícího paprsku. Toto naklonění je uskutečněné pomocí dvou modelářských servomotorů, které jsou ovládány počítačem přes IQRF.

Schéma zapojení IQRF a servomotorů je na obr. 19. Napájení (VCC) servomotorů je pomocí IQRF. GND je zapojeno do pinu č. 7 (GND). Signálové (žluté vodiče) jsou zapojeny do C1 a C5.



Obrázek 19: Blokové schéma připojení IQRF se servomotory, upraveno [4]

Programování tohoto pohybu je uskutečněno pomocí IQRF IDE4 a jazyka C. Vysílací modul IQRF je připojen k počítači s vývojovým prostředím. Servomotory jsou připojeny k přijímacímu modulu IQRF. Při vyslání příkazu „X + číslo“ servomotor otočí zrcátkem po směru osy X a při zadání příkazu „Y + číslo“ je zrcátko otočeno ve směru osy Y. Tyto příkazy jsou předprogramovány v makrech.

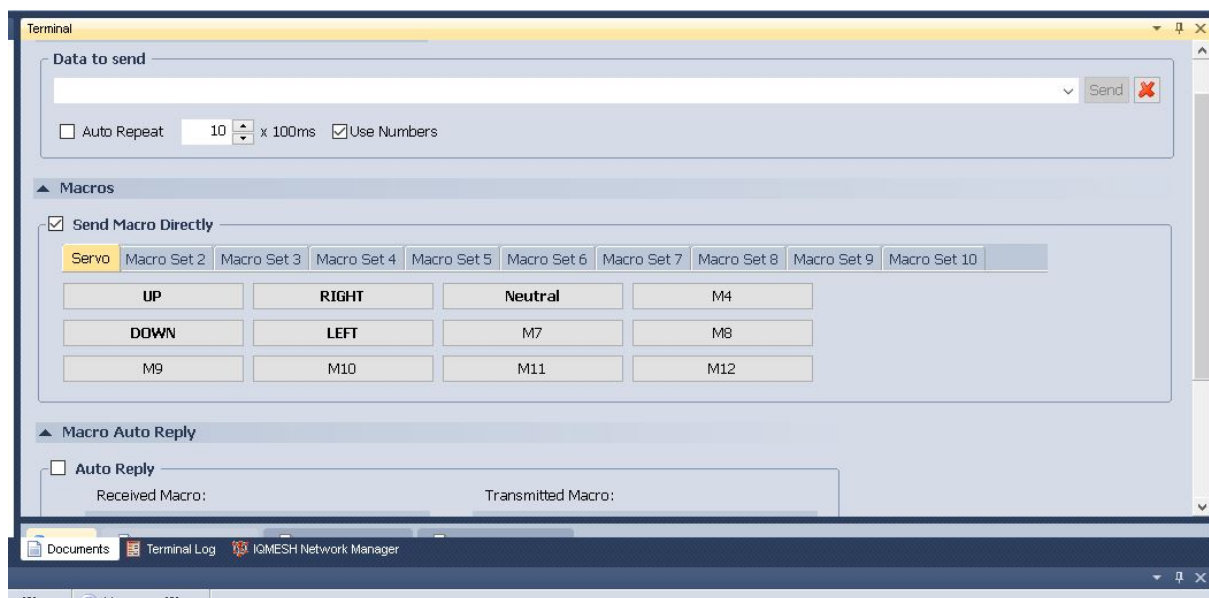
Níže je výpis kódu, který ovládá pohyb zrcátka po směru osy Y. Pro pohyb zrcátka ve směru X je potřeba nahradit Y písmenem X. Celý kód je zobrazen v příloze na CD.

```
// nastaveni poloh, pohybu
while (1)
{
    if (checkRF(RX_FILTER))
    {
        if (RFRXpacket())
        {
            pulseLEDR();
            if (bufferRF[0] == 'Y')
            {
                if (bufferRF[1] == INCREMENT)
                {
                    servo_pulse = servo_pulse + 1;
                    if (servo_pulse > HIGH_LIMIT)
                    {
                        servo_pulse = HIGH_LIMIT;
                    }
                }
                else if (bufferRF[1] == DECREMENT)
                {
                    servo_pulse = servo_pulse - 1;
                    if (servo_pulse < LOW_LIMIT)
                    {
                        servo_pulse = LOW_LIMIT;
                    }
                }
            }
            else
            {
                if (bufferRF[1] < LOW_LIMIT) servo_pulse = LOW_LIMIT;
                else if (bufferRF[1] > HIGH_LIMIT) servo_pulse = HIGH_LIMIT;
                else servo_pulse = bufferRF[1];
            }
        }
    }
}
```

Výpis 1: Ovládání serv pomocí IQRF modulu

GUI (uživatelské rozhraní) pro ovládání pohybu je na obr. 20. Tlačítka **UP** a **DOWN** pohybují servomotorem ve směru osy X a tlačítka **RIGHT** a **LEFT** pohybují se servomotory ve směru osy Y.

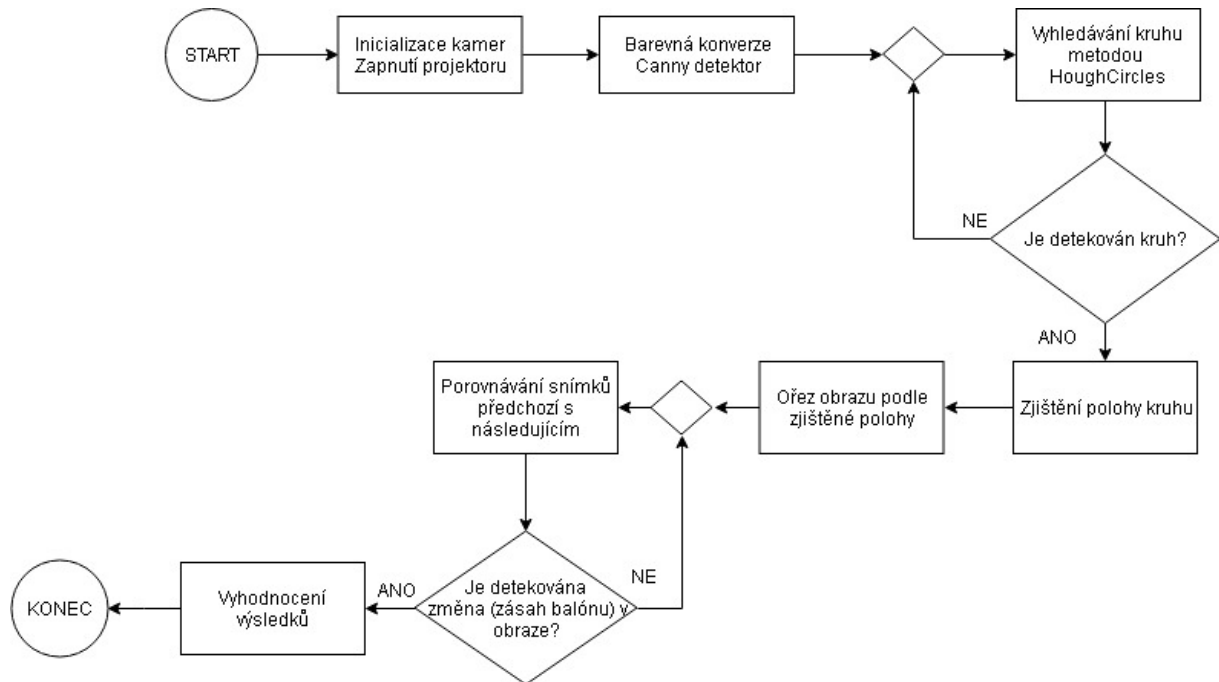
Tlačítko **NEUTRAL** nastaví servomotory do výchozí polohy.



Obrázek 20: Ukázka GUI na ovládání IQRG

7 Vytvoření uživatelského prostředí v jazyce C#

Pro vytvoření aplikace byl zvolen programovací jazyk C#, Windows Forms a knihovny EmguCV. Diagram této aplikace je na obrázku 21. Podrobný popis jednotlivých částí této aplikace je v následujícím textu. Celkový kód programu je v příloze na CD.



Obrázek 21: Diagram aplikace

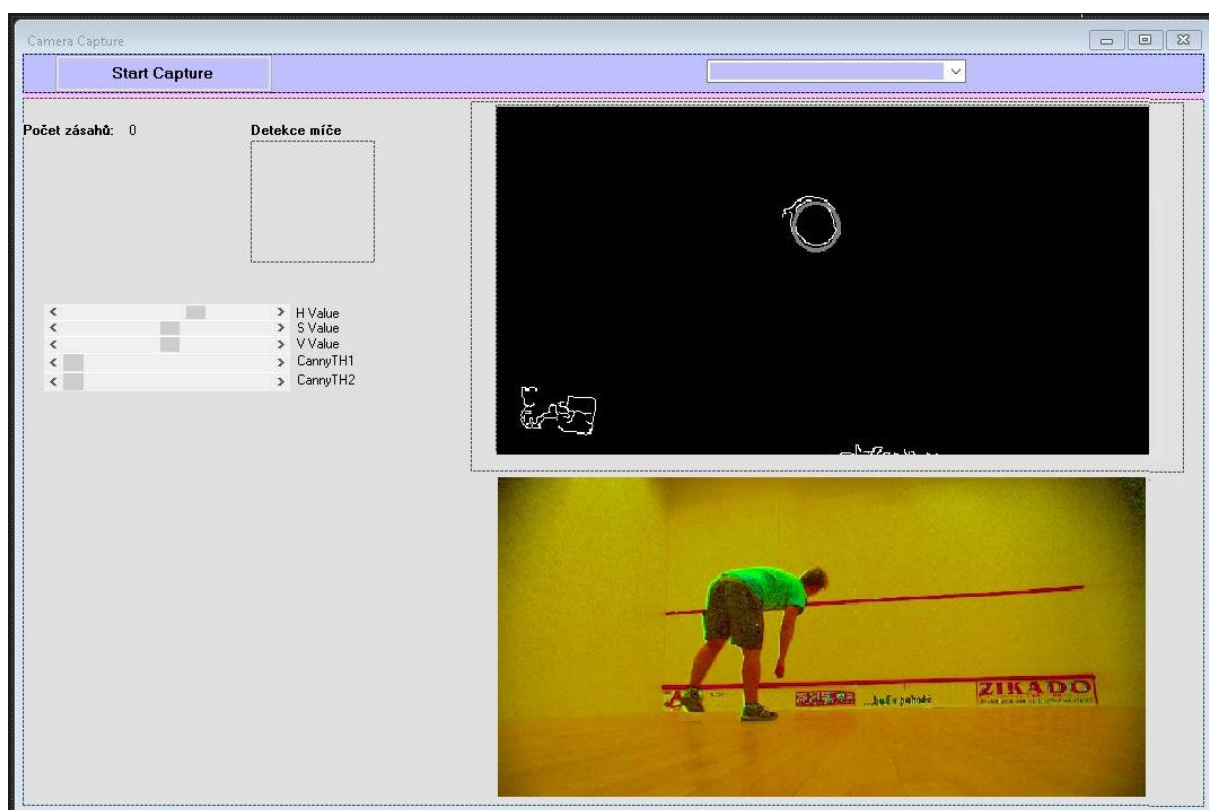
7.1 GUI

Uživatelské rozhraní (GUI) obsahuje tlačítko **Start Capture**, kterým se tato aplikace spustí. Po tomto spuštění se text tohoto tlačítka změní na **Stop** a je možno tímto tlačítkem aplikaci zastavit.

Dalšími prvky v GUI jsou **okna pro zobrazování videa z kamer**, **jedno vyhodnocovací okno** a **posuvníky** pro drobné doladění obrazu z kamer. Tyto posuvníky umožňují doladit výsledný obraz při změně např. světelnosti na kurtu. Je doporučeno toto doladění obrazu udělat před samotnou hrou.

Posledním prvkem v uživatelském prostředí je **počítadlo** úspěšných zásahů míčku v kruhu, které při detekci míčku v kruhu vždy přičte hodnotu.

Toto uživatelské rozhraní je ukázáno na obrázku 22.



Obrázek 22: Uživatelské rozhraní

7.2 Zahájení snímkování

Tlačítkem **Start Capture** se spustí celá aplikaci a tím se inicializují kamery a zahájí se snímkování z těchto kamer.

Aby byly v aplikaci jednotlivé videa z kamer rozlišeny je potřeba tyto kamery zaindexovat. Index [1] je pro kameru Basler a index [0] je pro akční kameru. Jednotlivé snímky (frame) z těchto kamer jsou uloženy do Matic.

Snímky z kamery Basler se ukládají do matice s názvem frame a snímky z akční kamery jsou ukládány do matice s názvem frame2.

Pro kameru Basler je nastavené rozlišení 2040 x 1086 pixelů a pro akční kameru je nastaveno rozlišení 960 x 540 pixelů.

Toto snímkování je naprogramováno viz následující výpis kódu.

```
public CameraCapture()
{
    InitializeComponent();
    CvInvoke.UseOpenCL = false;
    try
    {
        _capture = new VideoCapture(WebCams[1].Device_ID);
        _capture2 = new VideoCapture(WebCams[0].Device_ID);
        _capture.SetCaptureProperty(CapProp.FrameWidth, 2040);
        _capture.SetCaptureProperty(CapProp.FrameHeight, 1086);
        _capture2.SetCaptureProperty(CapProp.FrameWidth, 960);
        _capture2.SetCaptureProperty(CapProp.FrameHeight, 540);
        _capture.ImageGrabbed += ProcessFrame;
        _capture2.ImageGrabbed += ProcessFrame2;
    }
    catch (NullReferenceException excpt)
    {
        MessageBox.Show(excpt.Message);
    }
    _frame = new Mat();
    hsvimage = new Mat();
    _frame2 = new Mat();
}
```

Výpis 2: CameraCapture

7.3 Zpracování obrazu

Promítaný kruh, který hledáme, je na původních snímcích nevýrazný (viz obr. 23). Tato nevýraznost na snímcích je způsobená vysokou světelností na kurtech. Světelnost na kurtu se pohybuje okolo 1000–1500 Luxů. Tyto snímky jr proto potřeba upravit. Tyto úpravy jsou následující:

1. Převodění záznamu z RGB modelu do HSV modelu (viz obr. 24).
2. Jsou upraveny jednotlivé složky HSV modelu.
3. Po této úpravě následuje převedení zpět do RGB.
4. Snímek je překonvertován na monochromatický (viz obr. 25).
5. A jako poslední krok je použit Cannyho hranový detektor (viz obr. 26).

Výsledný snímek je již vhodný k vyhledávání kruhu a na obrázku 26 lze vidět tento kruh zvyrazněný.

Záznam z druhé, akční kamery, není nutno upravovat, protože viditelnost je uspokojivá a slouží pouze k zachycení pohybu člověka.



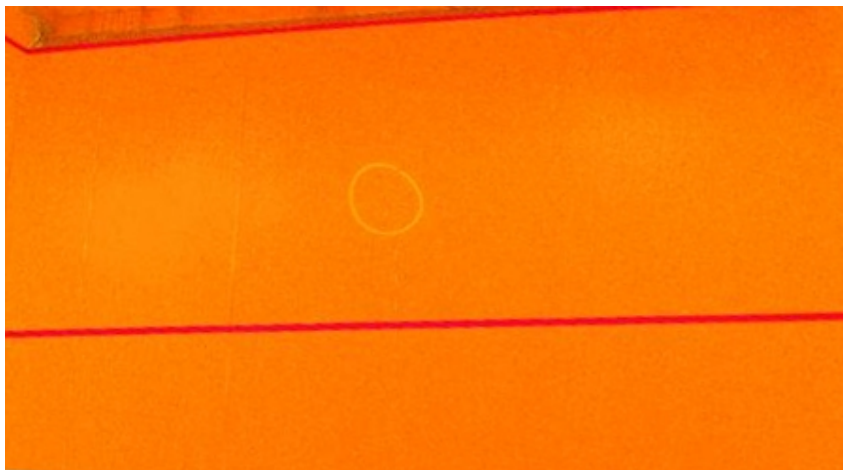
Obrázek 23: Snímek z kamery Basler bez úprav

7.3.1 Převod záznamu z RGB do HSV modelu

Tato konverze je uskutečněna pomocí metody `CvtColor` ze třídy `CvInvoke`. Obecně tato metoda převádí vstupní obrázek z jednoho barevného prostoru do jiného barevného prostoru. Parametry této třídy jsou *vstupní obrázek*, *výstupní obrázek* a *kód barevné konverze*. Kód pro převod z RGB do HSV je **Rgb2Hsv**.

7.3.2 Úprava jednotlivých složek modelu HSV

Jednotlivé složky jsou nastaveny pomocí posuvníku na GUI. Jmenovitě to jsou H Value (barevný tón), S Value (hodnota sytosti) a V Value (hodnota jasu). Tyto hodnoty nastavuje uživatel, či obsluha sama dle aktuálních podmínek (intenzita osvětlení, barevnost osvětlení a velikost prostoru). Výsledek, po této úpravě, by měl být podobný, jako na obr. 24.



Obrázek 24: Snímek z kamery Basler po převedení do HSV barevného modulu

7.3.3 Převedení do RGB a monochromatického zobrazení

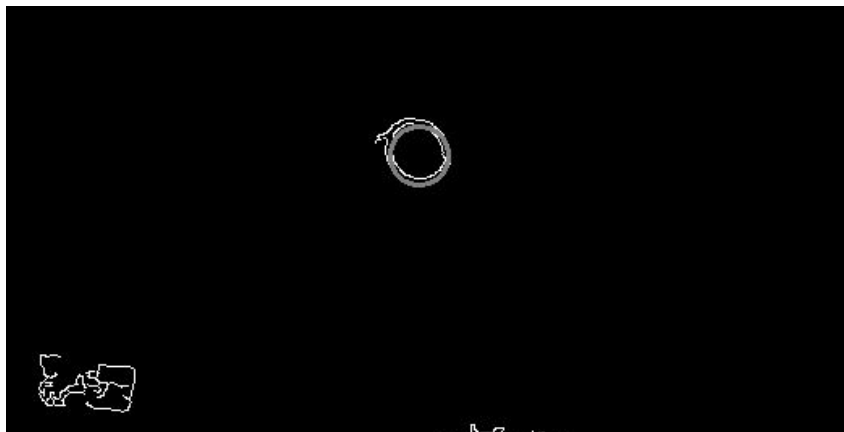
Následně je potřeba tento snímek pro další zpracování převést do monochromatického zobrazení. Toto zobrazení je nutné, kvůli následné aplikaci Cannyho detektoru. Výsledek je znázorněn na obrázku 25. Pro tuto konverzi se využívá metody obdobné jako pro převádění do HSV modulu, kód pro převod je označován jako **Rgb2Gray**.



Obrázek 25: Snímek z kamery Basler po převedení do HSV barevného modulu a následně převeden do monochromatického obrázku

7.3.4 Aplikace Cannyho detektoru

Tento detektor vyhledává okraje na vstupním obrázku a vyznačuje je v obrázku výstupním. Parametry této metody jsou *vstupní snímek*, *výstupní snímek*, *práh1* a *práh2*. Dále obsahuje dva volitelné parametry. Práh1 je práh minimální a práh2 je práh maximální. Tyto prahy umožňují přesnější eliminaci šumu z obrázku. Hodnoty těchto prahů si uživatel opět navolí dle aktuálních podmínek na uživatelském rozhraní u posuvníků CannyTH1 a CannyTH2. Výsledný snímek je na obrázku 26.



Obrázek 26: Snímek z kamery Basler po převedení do HSV barevného modulu, následně do monochromatického obrázku a aplikován Cannyho hranový detektor

7.4 Detekce kruhu a ořez vybrané oblasti

Aby bylo možno vyhodnotit, zda-li je úspěšný zásah míčku, je potřeba nejprve tento kruh vyhledat. Na toto vyhledávání je použita metoda `HoughCircles` ze třídy `CvInvoke`, kterou obsahuje knihovna `EmguCv`.

Vyhledávání kruhu probíhá v upraveném obrázku s aplikovaným Cannyho detektorem. Parametry této třídy jsou: *vstupní obrázek*, *metoda*, *inverzní poměr* a *minimální vzdálenost mezi středy zjištěných kruhů*. Další parametry jsou volitelné. Konkrétní hodnoty jsou 2 pro inverzní poměr, 2000 pro minimální vzdálenost mezi kruhy. Hodnota 80 udává minimální poloměr a hodnota 90 udává maximální poloměr. Tyto hodnoty byly zjištěny experimentálně.

Výsledný kruh se na zpracovaném obrazu obkreslí (viz obr. 26) a kolem tohoto kruhu se vytvoří čtvercový výřez. Tento výřez umožňuje výpočetně méně náročnou práci s výsledným obrazem a tím i rychlejší detekci míčku.

Výpis kódu této detekce je níže.

```

CircleF[] circles = CvInvoke.HoughCircles(Gray_HSV_Image, HoughType.Gradient, 2,
    2000, 80, 90);
foreach (CircleF circle in circles)
    CvInvoke.Circle(Gray_HSV_Image, Point.Round(circle.Center), (int)circle.Radius
        , new Bgr(Color.Purple).MCvScalar, 2);
    if (circles.Length > 0)
{
    this.crop = new Rectangle(new Point((int)(circles[0].Center.X - circles[0].
        Radius), (int)(circles[0].Center.Y - circles[0].Radius)), new Size((int)
        (100), (int)(100)));
}

```

Výpis 3: Vyhledání kruhu a vytvoření ořezu na základě informací o poloze kruhu

7.5 Vyhodnocení

Finální výřez má rozměr 100x100 a bere se podle souřadnic z obrázku v monochromatickém zobrazení. Z tohoto výřezu je zjišťováno, zda-li došlo k zásahu míčem. Tato detekce se provádí rozdílovou metodou, kde se porovnává snímek současný s předchozím. Pokud je rozdíl jednotlivých pixelů větší než 350, znamená to, že v kruhu byl detekován míč. Tato hodnota byla stanovena experimentálně, na základě několika pokusů. Daná metoda je z knihovny EmguCV. Na detekci je uživatel upozorněn pomocí štítku „Detekován míč“ a zároveň se tento úspěšný pokus započítá do počítadla. Aktuální skóre vidíme při průběhu hry v GUI.

```

private bool boxTest(Image<Gray,byte> lastImage)
{
    ComparisonFrame = PreviousFrame.AbsDiff(lastImage);
    ComparisonFrame = ComparisonFrame.ThresholdBinary(new Gray(50), new Gray
        (255));
    imageBox4.Image = ComparisonFrame;
    int[] differenceValue = ComparisonFrame.CountNonzero();
    if (differenceValue[0] > 350)
    {
        BallInTheBox = true;
    }
    else
    {
        BallInTheBox = false;
    }
    return true;
}

```

Výpis 4: Porovnávání snímku

8 Možné využití v rehabilitaci

Celá výše popsaná aplikace se dá upravit pro rehabilitační cvičení. Takovým rehabilitačním cvikem může být cvičení motoriky u pacientů s náhlou mozkovou příhodou, nebo při rozvoji motoriky u dětí.

8.1 Náhle mozkové příhody - fokální cerebrální ischemie

Fokální cerebrální ischemie způsobují uzavřené mozkové cévy, na základě kterého dochází ke snížení krevního zásobení mozku. Nejčastěji je zasažena část v mozku arteria cerebri media. S takovou ischemií se setká až 85 % osob. Další častou mozkovou příhodou je cerebrální krvácení.

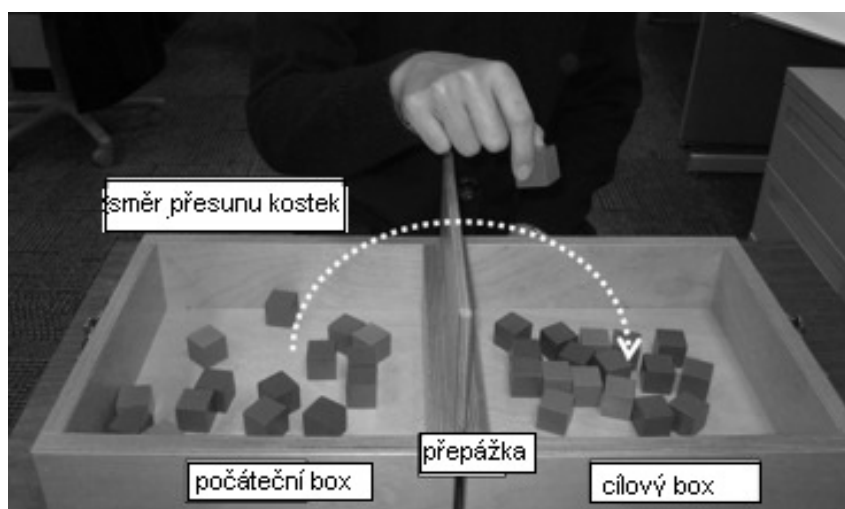
Pokud nedojde k včasnému zlepšení symptomatologie, je potřeba aplikovat kompenzační a regenerační mechanismy. Nejčastější deficity je uváděna jako senzomotorická paréza a její četnost výskytu je až 80 %. Na základě toho může dojít mimo jiné k ochrnutí horní končetiny a motoriky horní končetiny.

[12]

8.2 Rehabilitace po cévní mozkové příhodě

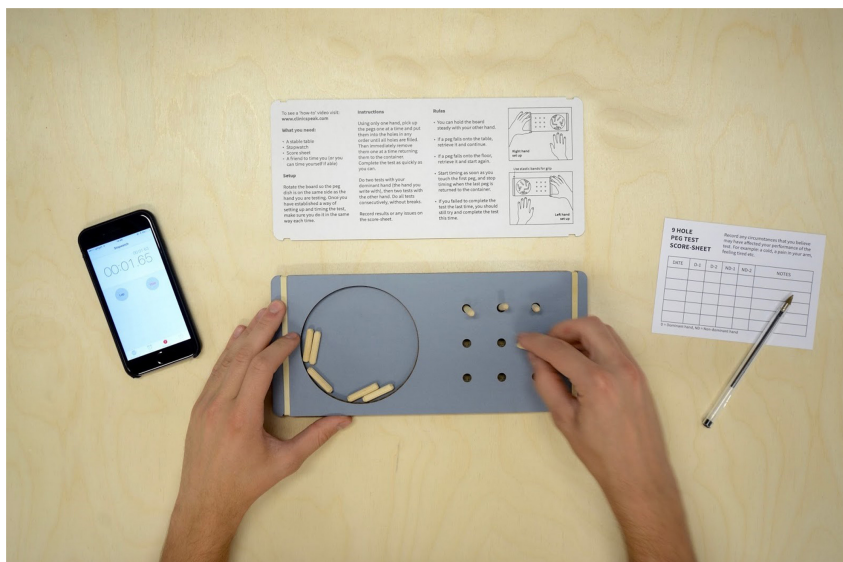
Metaanalýza prokazuje, že začátek rehabilitace během prvních sedmi dnů je neefektivnější. Musí být kontinuální. Pro rehabilitaci a pro posouzení stavu pacienta se používají různé testy.

Takovými testy jsou například **Box and Block Test** (obr. 27), kdy se vyšetřuje hrubá obratnost ruky za pomoci přesouvání kostek do krabice. Tento test je využívám i pro terapii horních končetin pro dospělé a děti s omezením obratnosti a motoriky.



Obrázek 27: Box and Block Test, upraveno dle [18]

Další test se nazývá **Nine Hole Peg Test** (obr. 28), tento test je podobný výše zmíněnému, zde je posuzována jemná motorika prstů. Pacient odebírá dřevěné tyčinky z misky a ty zapichuje do desky s otvory. Používá se pro pacienty s poruchami jemné motoriky a koordinace s lehkými parézami horních končetin a pro pacienty s roztroušenou sklerózou.



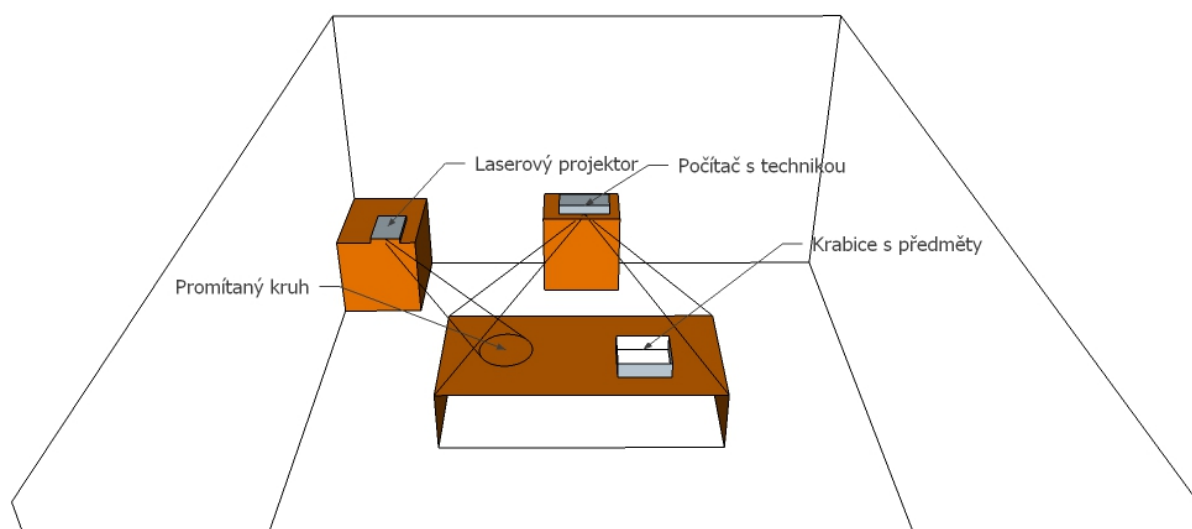
Obrázek 28: Nine Hole Peg Test [6]

Dalším testem je **Action Research Arm Test** v tomto testu pacient uchopuje různé předměty, které přenáší a manipuluje různými pohyby.
[12]

8.3 Vlastní návrh použití této aplikace pro rehabilitaci

Tato aplikace se dá využít i na podobné, výše zmíněné testy, a to například tak, že se promítá kruh na stůl a pacient přesouvá předměty do prostoru kruhu. Předměty mohou být umístěny v krabici, podobně jako při Box and Block Testu. Aplikace počítá, kolik předmětů je úspěšně přesunuto. Obtížnost se dá nastavovat zvětšením, nebo zmenšením poloměru kruhu. Nákres takového cvičení je na obr. 29.

[10]



Obrázek 29: Nákres pro rehabilitační cvičení po mozkové příhodě

9 Zhodnocení výsledků

V textu této bakalářské práce byla navržnuta a poté realizována aplikace pro squash. Tato aplikace byla naprogramována ve Windows Form v jazyce C# za použití funkcí z knihovny EmguCv. Tento program je schopen reagovat na změny v obraze a to v reálném čase.

Z pokusů bylo zjištěno, že k určení polohy předmětu, postačí jedna kamera. Touto kamerou je vysokorychlostní průmyslová kamera, která dokáže zachytit barevný obraz se snímkovací frekvencí až 165 fps. Druhá kamera sleduje pohyb hráče/pacienta a na základě tohoto videa je možno poskytnout zpětnou vazbu.

Dle intenzity osvětlení na kurtu, která se pohybuje okolo cca 1300 Luxů, bylo zjištěno, že pro označování místa, nelze využít běžně dostupných projektorů. Proto bylo nutno vytvořit vlastní laserový projektor. Kruh promítaný tímto projektorem již je při světelnosti 1000–1500 Luxů na stěně squashového kurtu viditelný.

Celá aplikace má za cíl vylepšení techniky a zlepšení přesnosti míření pro hráče squash. Je možno sledovat pohyb hráče, a na základě záznamu má hráč, či trenér zpětnou vazbu. Dále je možno sledovat vzestup hráče, díky nastavování obtížnosti průměrem kruhu.

Dále zde bylo ukázáno, jak lze tento program aplikovat pro rehabilitační účely, jako je posuzování stavu poškození motoriky po cévní mozkové příhodě.

10 Závěr

Cílem této bakalářské práce měla být aplikace, která detekuje předmět za pomoci dvou vhodně umístěných kamer v reálném čase. Po detekci předmětu je určena jeho přesná poloha na ploše. Výsledná aplikace, dokáže zpracovat obraz, vyhledat kruh z projektoru a určit tuto polohu. Dále dokáže reagovat na změny v obraze v reálném čase, což je využito pro detekci míčku.

První část bakalářské práce je zaměřená na shrnutí základních formátů pro ukládání obrázků, takovými formáty můžou být BMP, RAW, JPEG a další. Jsou zde vysvětleny základní barevné modely a jak se od sebe liší a základní metody úprav obrazu. Jsou popsány převody z barevného obrazu do monochromatického zobrazení, segmentace obrazu prahováním, detekce hran a jasové transformace. V další části je řešeno metod týkající se vyhledávání objektu v obrazu. Takovými metodami mohou být ROI metoda v kombinaci s PN algoritmem, Meanshift, CamShift, barevná extrakce, metoda odčítání pozadí a jiné.

Pro práci jsou využity dvě kamery. Jedna vysokorychlostní průmyslová a druhá je obyčejná akční kamera. Jsou zde popsány hlavní rozdíly v čípech pro zaznamenání obrazu v kamerách. Takové čipy jsou CCD a CMOS.

Praktická část práce zahrnuje funkční aplikaci na sledování progresu pacienta a možnost tréninku přesnosti zásahu při hře squash za pomoci nastavení průměru promítaného kruhu, do kterého se má sportovec za úkol trefit.

Další možné využití této aplikaci je při posuzování deficitu při poškození nervového systému. Je navrhnout test, který na základě výsledků určuje míru deficitu po cévní mozkové příhodě.

Literatura

- [1] *BHUSHAN Nemade. R. R. SEDAMAKER. Adaptive Automatic Tracking, Learning and Detection of Real-time Objects in the Video Stream.* IJAIS Proceedings on International Conference and workshop on Advanced Computing 2013.
- [2] *ČÁSTKA, Vítězslav. CPU Fan Laser Ring Projector.* [online]. [cit. 2018-03-23]. Dostupné z: <https://www.thingiverse.com/thing:2836052>.
- [3] *Datasheet. Basler.* [online]. [cit. 2017-11-15]. Dostupné z: <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca2000-165uc/>.
- [4] *Datasheet. IQRF Technical Guide.* [online]. [cit. 2017-12-27]. Dostupné z: <http://www.iqrf.org/weben/downloads.php?id=79>.
- [5] *DOBEŠ, Michal. Zpracování obrazu a algoritmy v C#.* Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-233-6.
- [6] *DUBUISSON, Nicolas. Angelika BAUER. Mark BUCKLEY. Rich GILBERT. Adam PATERSON. Monica MARTA. Sharmilee GNANAPAVAN. Benjamin TURNER. David BAKER. Gavin GIOVANNONI. Klaus SCHMIERER. Alison THOMSON. Validation of an environmentally-friendly and affordable cardboard 9-hole peg test.* Multiple Sclerosis and Related Disorders. DOI: <https://doi.org/10.1016/j.msard.2017.08.002>. <http://www.sciencedirect.com/science/article/pii/S2211034817301852>.
- [7] *HLAVÁČ, Václav. Milan SEDLÁČEK. Zpracování signálu a obrazu.* Elektrotechnická fakulta ČVUT v Praze. Praha. 1999.
- [8] *HORÁK, Karel. Ilona KALOVÁ. Petr PETYOVSKÝ. Miloslav RICHTER. Počítačové vidění.* Vysoké učení v Brně. Brno. 2008.
- [9] *HSIA, Chih-Hsien. Liou YUN-JUNG. Chiang JEN-SHIUN. Directional Prediction CamShift algorithm based on Adaptive Search Pattern for moving object tracking.* Journal of Real-Time Image Processing. DOI: 10.1007/s11554-013-0382-x. ISSN 1861-8219.
- [10] *CHOCHOLOUŠKOVÁ, Růžena. Metodika Rehabilitačního vzdělávacího programu: Svazek č. 4 - hrubá motorika.* Pardubice.
- [11] *LABAJ, Tomáš. Detekce křivek v obraze.* diplomová práce. Vysoké učení technické v Brně. Brno. 2009. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=116823.
- [12] *LIPPERTOVÁ-GRÜNEROVÁ, Marcela. Rehabilitace po náhlé cévní mozkové příhodě.* Praha: Galén, 2015. ISBN 978-80-7492-225-1.

- [13] *MICROSOFT. About Bitmaps.* [online]. [cit. 2018-03-24]. Dostupné z: <https://msdn.microsoft.com/en-us/library/windows/desktop/dd162461>.
- [14] *MONICA, V.M. K.G.J. NIGEL 2Object tracking based on Kalman filter and gait feature extraction* 2017. International Conference on Inventive Systems and Control (ICISC).
- [15] *MULYA, Anton Kurniawan. Fernando ARDILLA. Dadet PRAMADIHANTO. Ball tracking and goal detection for middle size soccer robot using omnidirectional camera.* In: 2016 International Electronics Symposium (IES) [online]. IEEE, 2016, 2016, s. 432-437 [cit. 2018-04-22]. DOI: 10.1109/ELECSYM.2016.7861045. ISBN 978-1-5090-1640-2. Dostupné z: <http://ieeexplore.ieee.org/document/7861045>.
- [16] *OpenCV. Meanshift and Camshift.* [online]. [cit. 2018-04-17]. Dostupné z: https://docs.opencv.org/3.1.0/db/df8/tutorial/_py/_meanshift.html.
- [17] *ROSEBROCK, Adrian. OpenCV shape detection.* [online]. [cit. 2017-12-30]. Dostupné z: <https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection>.
- [18] *SLOTA P. Gregory. Leah R. ENDERS. Na Jin SEO. Improvement of hand function using different surfaces and identification of difficult movement post stroke in the Box and Block Test.* Applied Ergonomics. DOI: <https://doi.org/10.1016/j.apergo.2013.10.014>. <http://www.sciencedirect.com/science/article/pii/S000368701300224X>.
- [19] *SOJKA, Eduard. Digitální zpracování a analýza obrazů.* VŠB-Technická univerzita-hru. 2000. ISBN 80-7078-746-5.
- [20] *ŠPANĚL, Michal. Vítězslav BERAN. Obrazové segmentační techniky.* Brno. Česká republika.[cit.2018-04-22]. Dostupné z: <http://www.fit.vutbr.cz/~spanel/segmentace/.cs.iso-8859-2>.
- [21] *ŠURKALA, Milan. Fotomobily: snímací čipy CMOS vs. CCD.* [online]. [cit. 2017-12-30]. Dostupné z: <https://www.digimanie.cz/fotomobily-snimaci-cipy-cmos-vs-ccd/2885>.

Přílohy na CD

- I. STA0319 bakalarska prace
- II. STA0319 Zdrojove kody programu
 - a. STA0319 program k BP
 - b. STA0319 ovladani serv